IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

We will predict if the Falcon 9 first stage will land successfully.

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Will SpaceX be successful?

- What are the main factors, that the rocket will land successful?
- What are the relationship between the variables and how can we affect or predict them?
- What are the best conditions for the rocket launch?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Develop SpaceX Rest API and web scraping from different (open) data sources
- Perform data wrangling
  - Data cleaning and one hot encoding data fields for processing in Machine Learning
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection – SpaceX API

GitHub:
[Data Collection API]

Ge Response from API

Create dataframe

Clean data with custom functions and create dictionary

Filter dataframe

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
response.json()
df=pd.json_normalize(response.json())
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

Finally lets construct our dataset using the data we have obtained. We we combine the colun

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9=data[data['BoosterVersion']!='Falcon 1']
data_falcon9.head()
```

Data Collection - Scrapping

- Ge Response from HTML
- Create BeautifulSoup Object
- Find Tables
- Get column names
- Create dictionary andAppend data to keys
- Converting dictionary to dataframe

```
r=requests.get(static_url)
```

```
soup=BeautifulSoup(r.text, 'html.parser')
```

```
html_tables = soup.find_all('table')
html_tables
```

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
        name = extract_column_from_header(temp[x])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

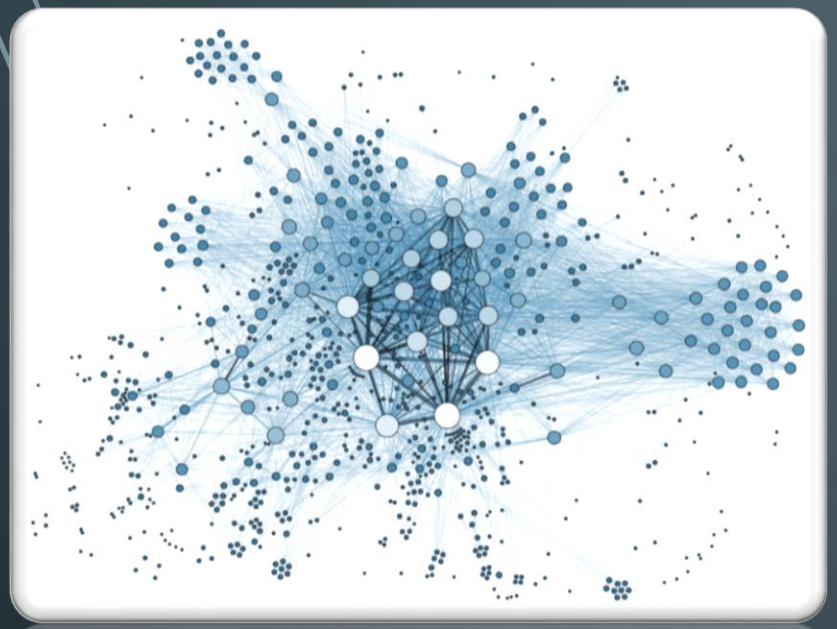
```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
```

```
df=pd.DataFrame(launch_dict)
df.head()
```

- GitHub:

[WebSrapping]

Data Wrangling



- GitHub: Data Wrangling

Calculate the number of launches at each side

```
df.value_counts('LaunchSite')
```

Calculate the number and occurence of each orbit

```
df.value_counts('Orbit')
```

Calculate the number and occurrence of mission outcome perorbit type

```
landing_outcomes = df.value_counts('Outcome')
landing_outcomes
```

Create a landing outcome label from Outcome Column

```
landing_class = []
for key,value in df['Outcome'].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

Calculate the success rate for landings

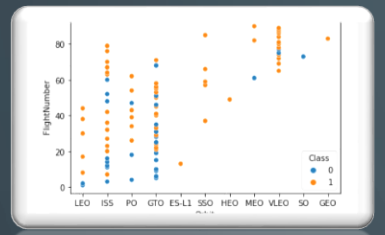
```
df["Class"].mean()
```

```
0.6666666666666666
```

Export DataSet

```
df.to_csv("dataset_part\_2.csv", index=False)
```

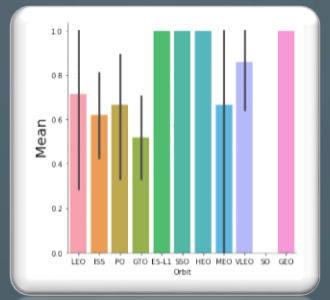
EDA WITH DATA VISUALIZATION



SCATTER GRAPHS

- Flight Number vs. Payload Mass
- Flight Number vs. Laucnch Site
- Payload vs. Launch Site
- Orbit vs. Flight Number
- Payload vs. Orbit Type
- Orbit vs. Payload Mass

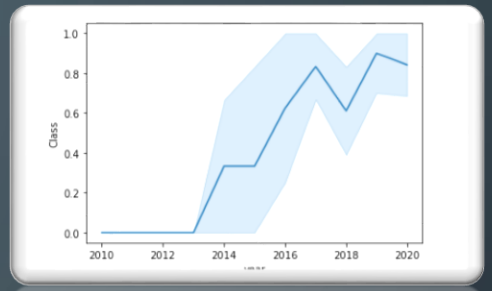
A scatter plot (also called a scatterplot, scatter graph, scatter chart, scattergram, or scatter diagram) is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. If the points are coded (color/shape/size), one additional variable can be displayed



BAR GRAPHS

- Mean vs. Orbit

A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a column chart.



LINE GRAPHS

- Success Rate vs. Year

A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a column chart.

GitHub: EDA DATA ViSUALIZATION

EDA with SQL

GitHub: EDA with SQL

- Display the names of the unique launch sites in the space mission
  - %sql select distinct(LAUNCH_SITE) from SPACEXTBL
- Display 5 records where launch sites begin with the string 'CCA'
  - %sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
- Display the total payload mass carried by boosters launched by NASA (CRS)
  - %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
- Display average payload mass carried by booster version F9 v1.1
  - %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
- List the date when the first successful landing outcome in ground pad was acheived.
  - %sql select min(DATE) from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)'
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - %sql select BOOSTER_VERSION from SPACEXTBL where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
- List the total number of successful and failure mission outcomes
  - %sql select count(MISSION_Outcome) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - %sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - %sql select BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where LANDING__OUTCOME = 'Failure (drone ship)'
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
  - %sql select * from SPACEXTBL where LANDING__OUTCOME like 'Failure (drone ship)' or LANDING__OUTCOME like 'Success (ground pad)' and (DATE between '2010-06-04' and '2017-03-20') order by date desc

Build an Interactive Map with Folium

- To visualize the launch sites line an interactive map at latitude and longitude at each side a circle marker with a label was added.
- Then a marker cluster was added where green markaer indicate a successful launch and a red marker indicates a failure. Marker cluster is a good option for a clear arrangement.
- In a third step some PolyLines were added to visualize the distance to some landmarks like railways, highways, coastline or cities.

- GitHub: Interactive Map with Folium

Build a Dashboard with Plotly Dash

- First of all pie chart was implemented that are showing the total launches by a certain site or summarize the results of all sites. You can see the success count for the sites. You can select the sites with a drop down menu.
- The second plot shows a scatter graph with the relationship with outcome, Payload Mass, an different booster versions according to the selected launch site

GitHub: Plotly Dash

Predictive Analysis (Classification)

BUILD

- Load and transform data
- Split the dataset inti training and testing data
- Choose the ML algorithm

EVALUATE

- Check accuracy for each model
- Plot confusion matrix

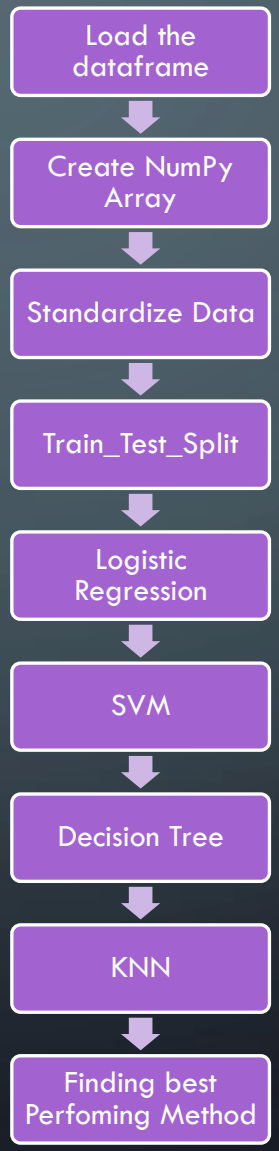
IMPROVE

- Feature engineering
- Tuning the algorithm

DECISION

- The model with the highest score due to accuracy was chosen

- GitHub: predictive analysis lab



Results

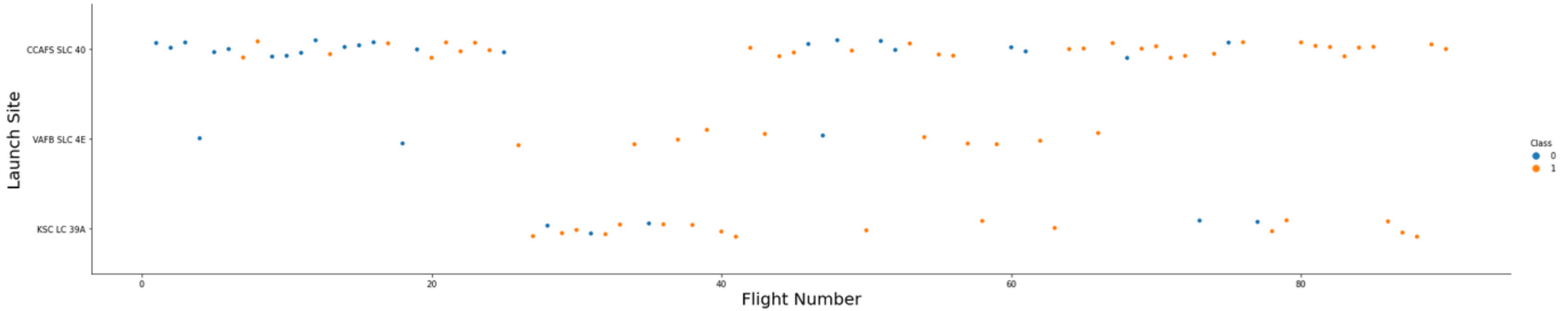
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

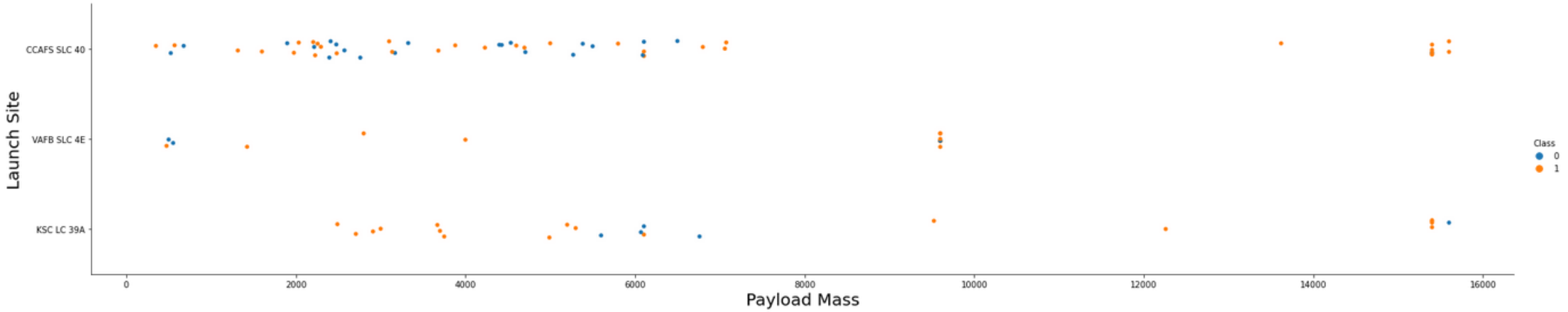
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



With higher flight numbers the success rates increases

Payload vs. Launch Site

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

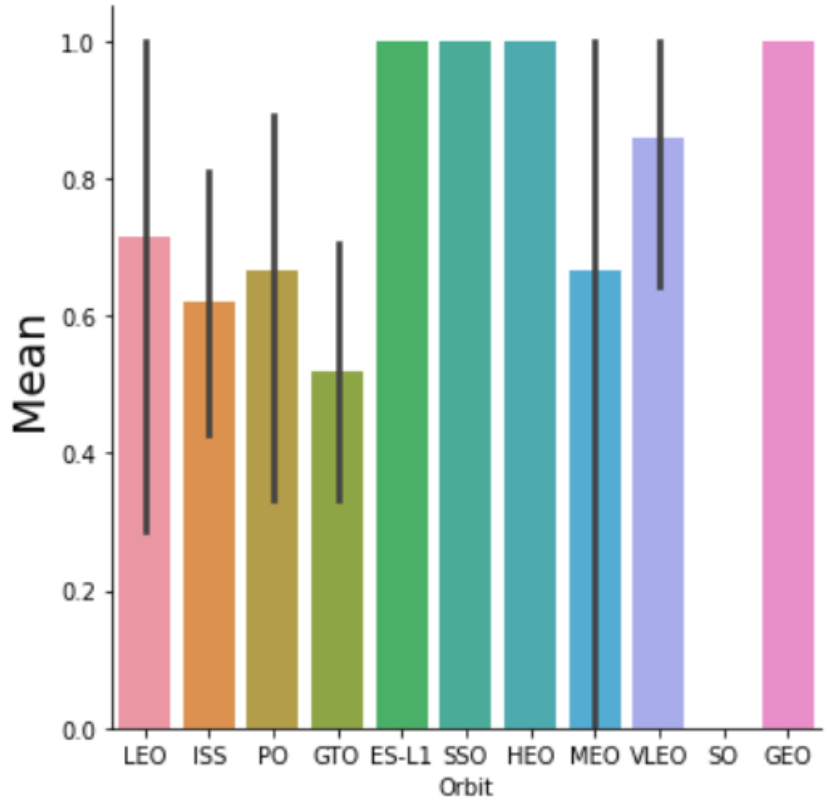


- for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000)
- The success rate for higher payloads is higher than for lower payloads

Success Rate vs. Orbit Type

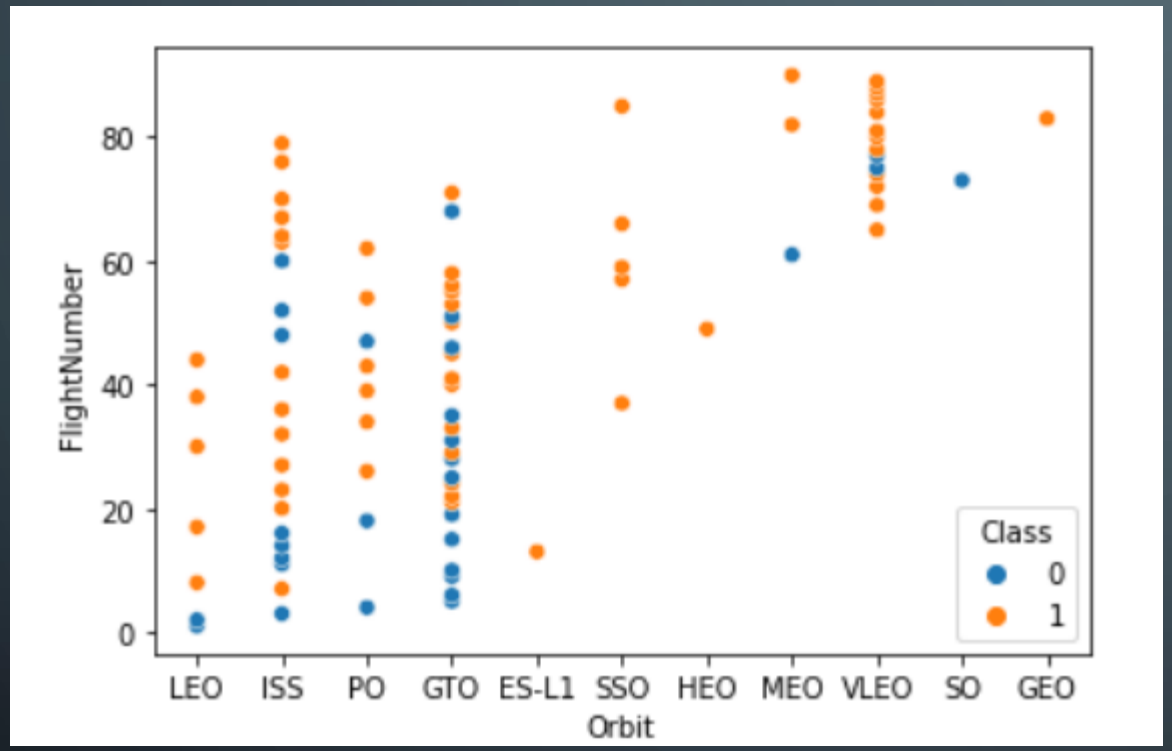
```
sns.catplot(x='Orbit', y='Class', kind='bar', data=
plt.ylabel('Mean', fontsize=20)
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=
```



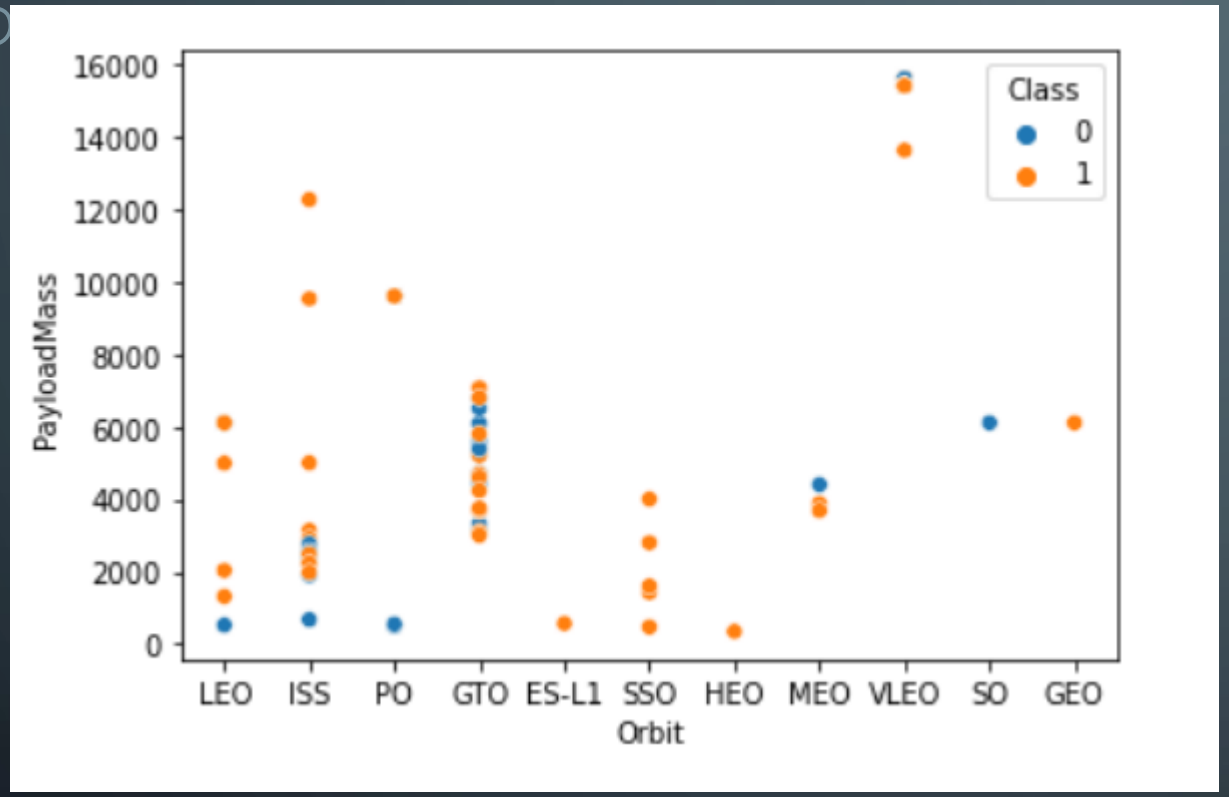
- GEO, HEO, SSO and ES-L1 has a 100% success rate

Flight Number vs. Orbit Type



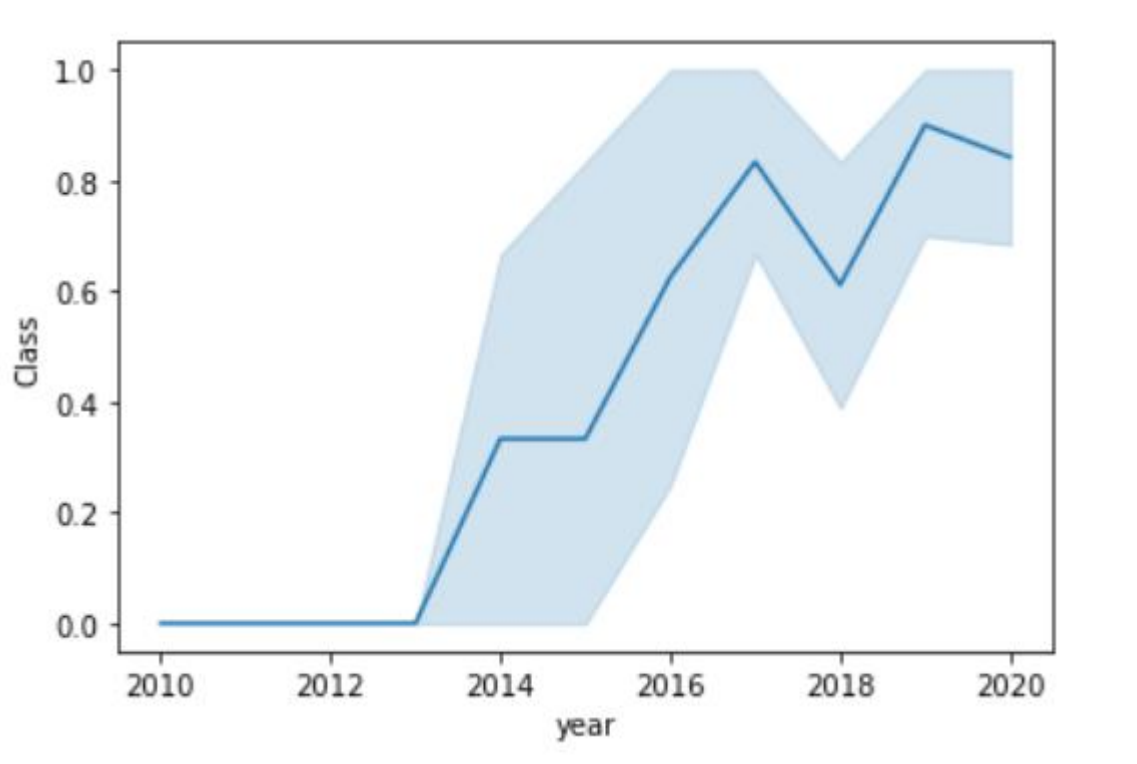
- LEO orbit the Success appears related to the number of flights;
- There seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- For GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



- you can observe that the success rate since 2013 kept increasing till 2020
- In 2018 the success rate decreases

All Launch Site Names

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

```
 * ibm_db_sa://dsc97143:***@55fbc997-9266-4331-afd3-8
/bludb
Done.
```

launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

The expression DISTINCT will giv you the unique values in column „Launch_site“ from table SPACEXTBL

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

```
 * ibm_db_sa://dsc97143:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929
/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Wizh teh expression „like ‘CCA%‘“ you get the launch sites that starts with cca; the percentage in the expression ensure, that the value starts with cca.

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://dsc97143:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od
/bludb
Done.
```

1
45596

The expression sum gives you the sum of the column „PAYLOAD_MASS“ and the expression where filters the expressions to the customer NASA (CRS)

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://dsc97143:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.datab
/bludb
Done.
```

1
2928

The expression avg gives you the average of the column „PAYLOAD_MASS_KG“

First Successful Ground Landing Date

```
%sql select min(DATE) from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://dsc97143:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.dat
/bludb
Done.
```

1
2015-12-22

The expression min gives you the lowest value of the column „DATE“

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select BOOSTER_VERSION from SPACEXTBL where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000
and PAYLOAD_MASS__KG_ < 6000
```

```
 * ibm_db_sa://dsc97143:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929
/bludb
Done.
```

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

The Expression whre filters the data set to Landing oztcome and payload mass

Total Number of Successful and Failure Mission Outcomes

```
%sql select count(MISSION_Outcome) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in f
light)'
```

```
 * ibm_db_sa://dsc97143:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929
/bludb
Done.
```

1
100

The expression count counts the value withe the following criterias

Boosters Carried Maximum Payload

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * ibm_db_sa://dsc97143:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929
/bludb
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

There we used a subquery to get the maximum payload mass

2015 Launch Records

```
%sql select BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where LANDING__OUTCOME = 'Failure (drone ship)' and DATE like '2015%'
```

```
 * ibm_db_sa://dsc97143:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

The dataset is filtered to „Failure (drone ship)“ in 2015

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

```
%sql select * from SPACEXTBL where LANDING__OUTCOME like 'Failure (drone ship)' or LANDING__OUTCOME like 'Success (grou
nd pad)' and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```

```
 * ibm_db_sa://dsc97143:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929
/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-06-15	14:29:00	F9 FT B1024	CCAFS LC-40	ABS-2A Eutelsat 117 West B	3600	GTO	ABS Eutelsat	Success	Failure (drone ship)
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-01-17	18:42:00	F9 v1.1 B1017	VAFB SLC-4E	Jason-3	553	LEO	NASA (LSP) NOAA CNES	Success	Failure (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-01-10	09:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)

The dataset is filtered according to the statements.
The last expression „order“ ordered the dataset in a descending order due to date

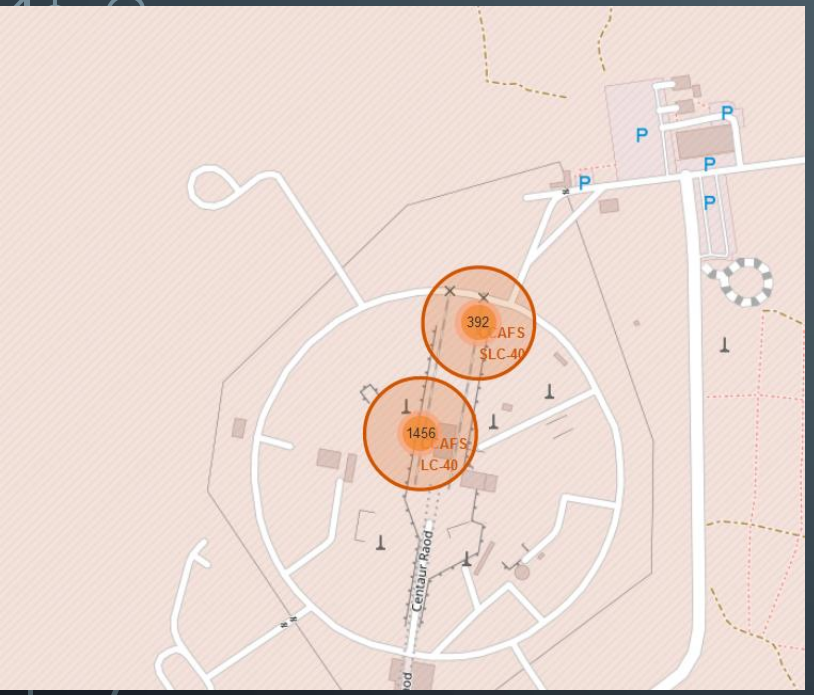
Section 4

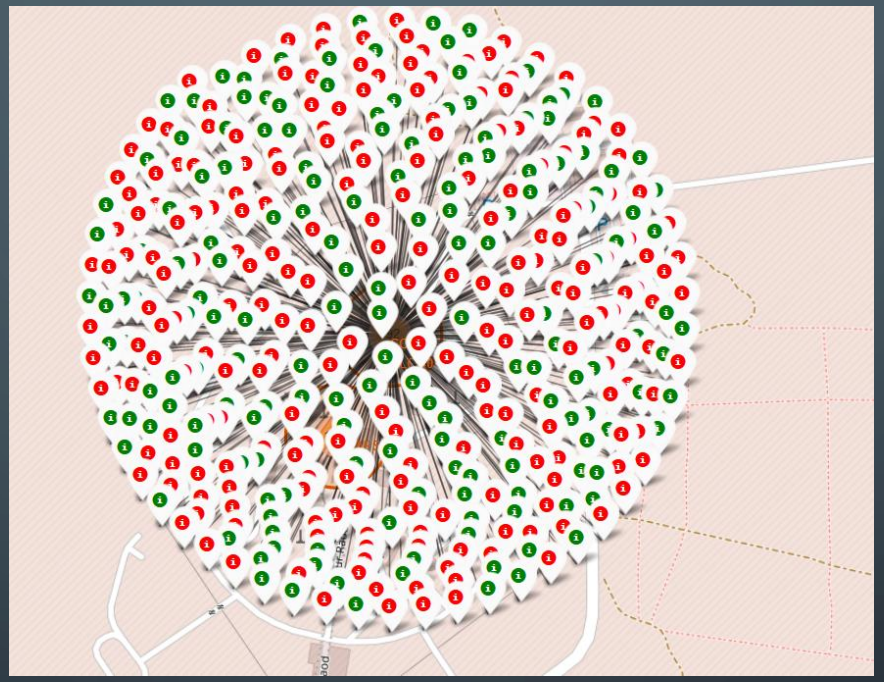
Launch Sites Proximities Analysis

All Launch sites are close to the Equator

All launch sites are located in the U:S close to the coast

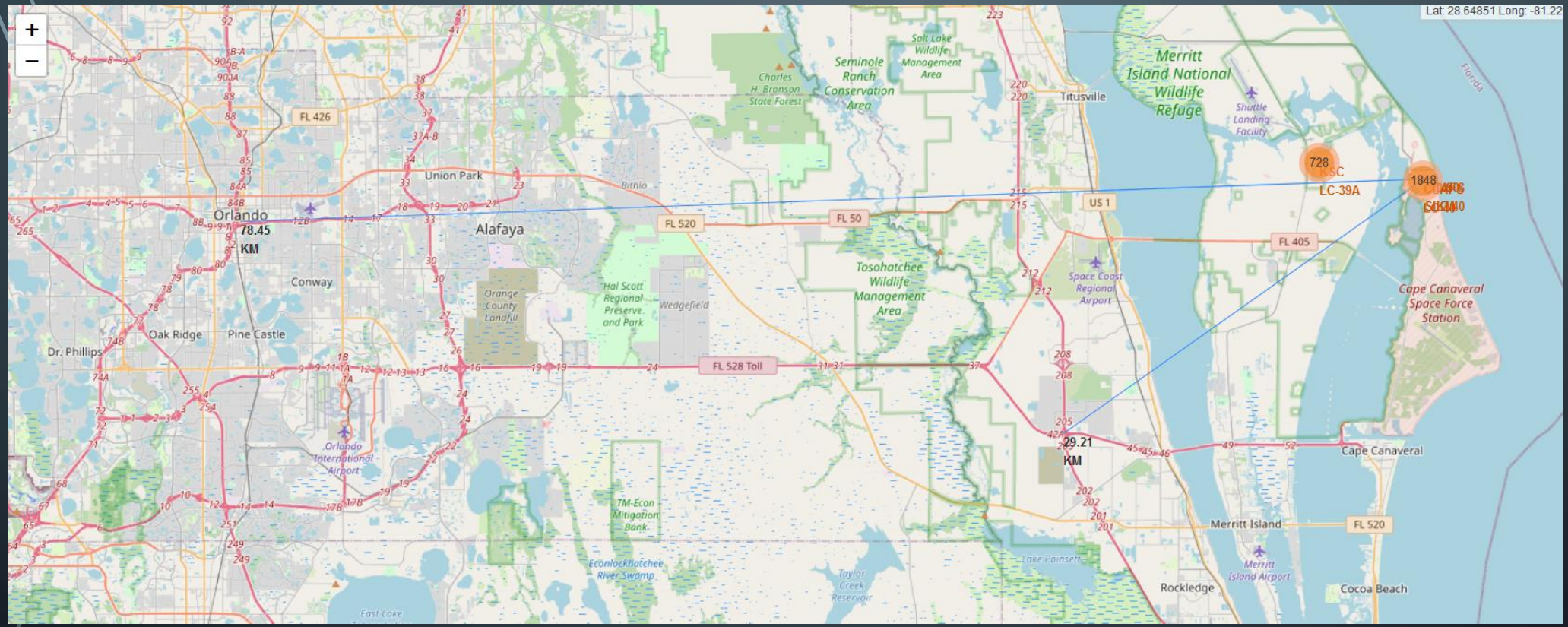
Florida Launch Sides



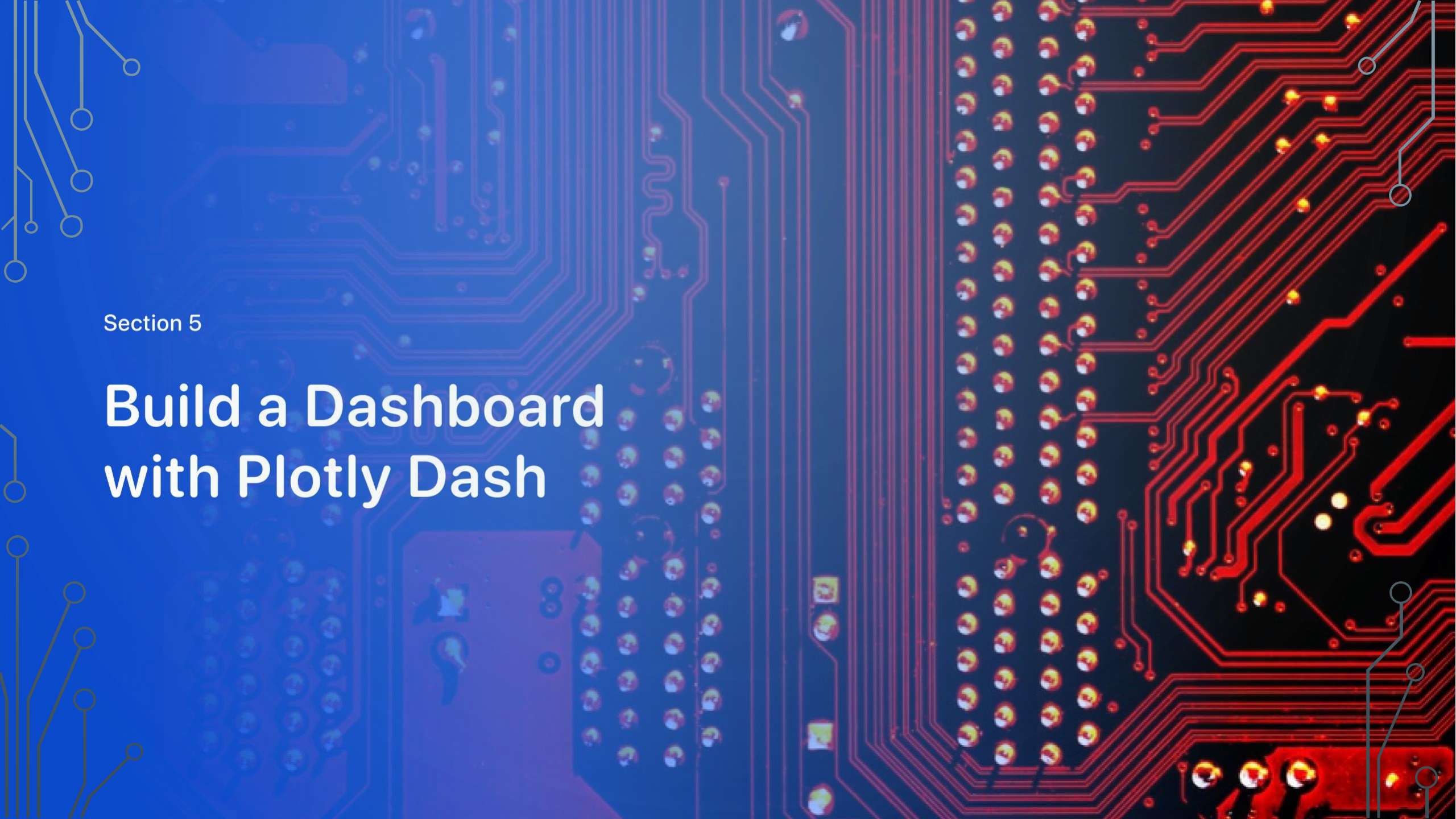


You can see with green markers the successful and with red markers failed launches

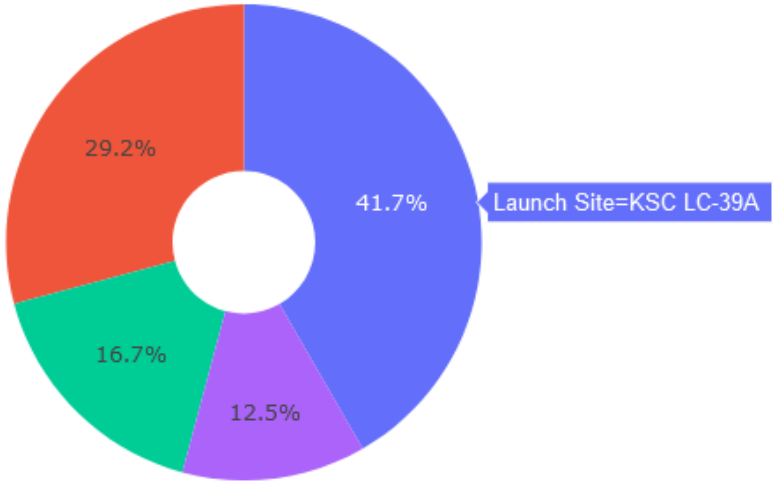
Distances to Landmarks



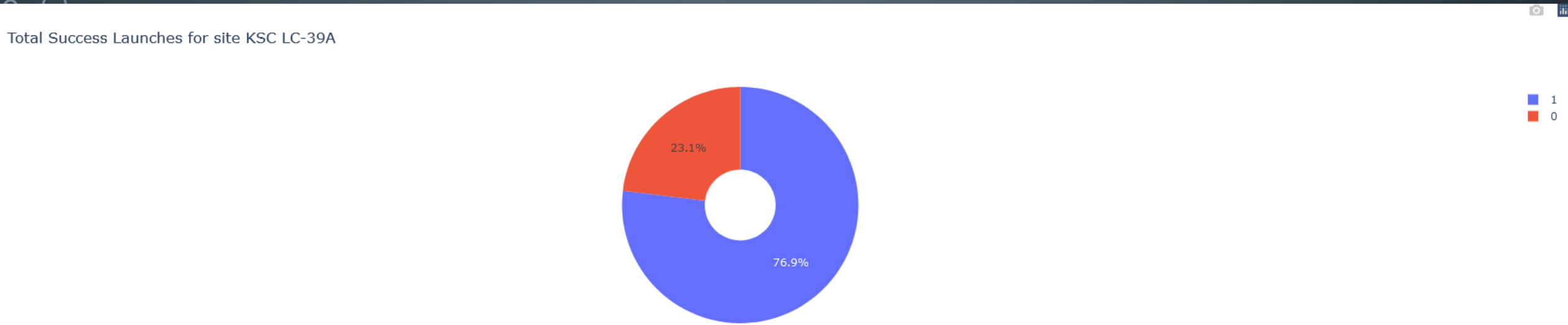
You can see the distances to landmarks like Center of Orlando, Railways, Highways and Coast


Section 5
Build a Dashboard with Plotly Dash

KSC LC-39A has the highest Success rate



KSC LC-39A has the highest success ratio



The success rates for low payload mass is higher than for high ones

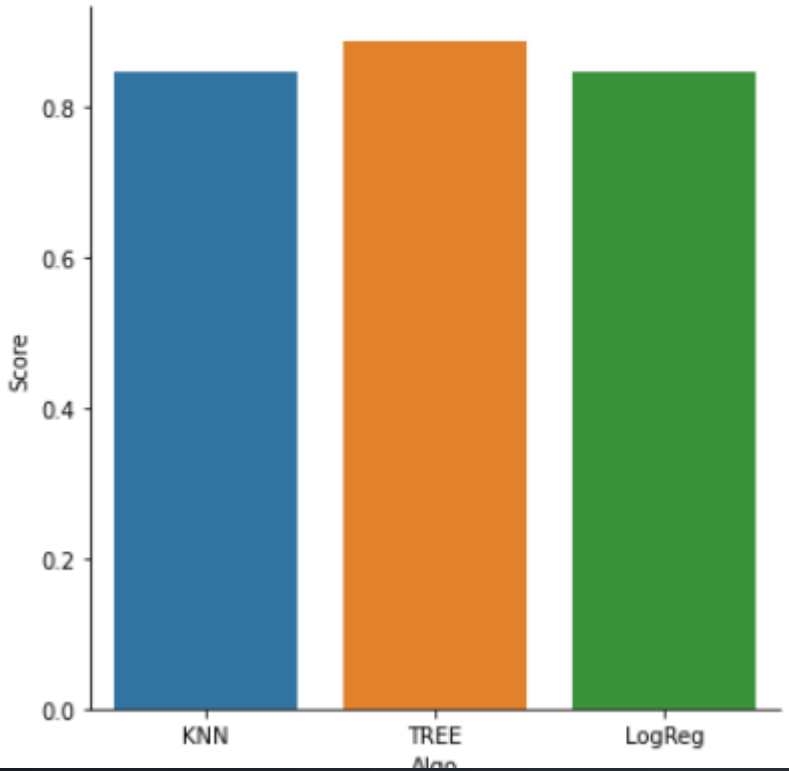
Section 6

Predictive Analysis (Classification)

Classification Accuracy

```
sns.catplot(x= 'Algo',y= 'Score', kind='bar', data=df)
```

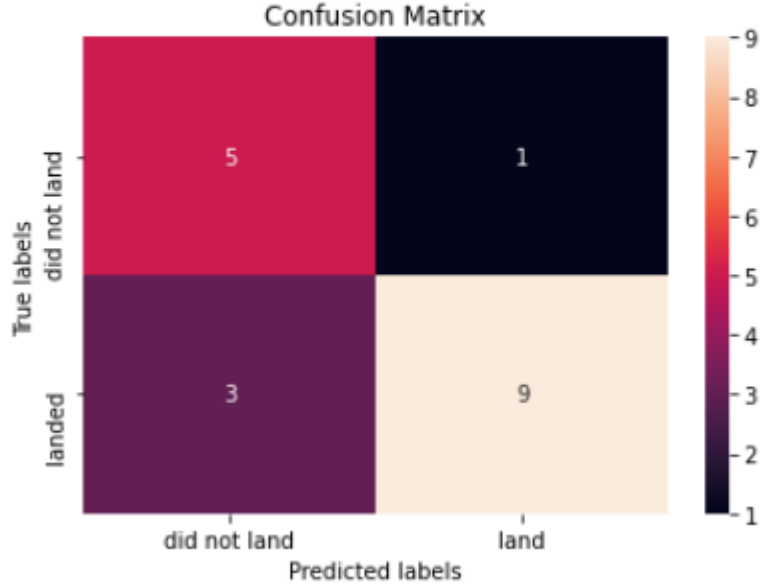
```
]: <seaborn.axisgrid.FacetGrid at 0x7f9d723fe790>
```



The TREE-Algorithm has the highest value

Confusion Matrix

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



The main problem of this algorithm is a false negative prediction for landed boosters

Conclusions

- Low weight payload performs better
- High flight numbers perform better → learning curve
- KSC LC39A is the best performing launch site
- GEO, HEO, SSO and ES-L1 missions are the best performing launches
- The Tree Algorithm is the best performing prediction model

Thank you!